

Assignemtn No 5

DC

Code:

Hostel_server.py

```
import Pyro4
```

```
@Pyro4.expose
```

```
class HotelBooking:
```

```
    def __init__(self):
```

```
        self.bookings = {} # Stores guest_name -> room_number
```

```
    def book_room(self, guest_name):
```

```
        if guest_name in self.bookings:
```

```
            return f"Guest '{guest_name}' already has a booking in Room
```

```
{self.bookings[guest_name]}"
```

```
        room_number = len(self.bookings) + 1
```

```
        self.bookings[guest_name] = room_number
```

```
        return f"Room {room_number} booked for '{guest_name}'"
```

```
    def cancel_booking(self, guest_name):
```

```
        if guest_name not in self.bookings:
```

```
            return f"No booking found for '{guest_name}'"
```

```
        del self.bookings[guest_name]
```

```
        return f"Booking canceled for '{guest_name}'"
```

```
# Start the Pyro4 daemon and register the object
```

```
def main():
```

```
    daemon = Pyro4.Daemon() # Start a Pyro daemon
```

```
    ns = Pyro4.locateNS() # Locate the name server
```

```
    hotel_booking = HotelBooking() # Instantiate the class
```

```
    uri = daemon.register(hotel_booking) # Register the object instance
```

```
    ns.register("hotel.booking", uri) # Register with a unique name
```

```
    print("Hotel Booking Server is running...")
```

```
    daemon.requestLoop() # Keep server running
```

```
if __name__ == "__main__":
```

```
    main()
```

Hostel_client.py

import Pyro4

Connect to the remote object

hotel = Pyro4.Proxy("PYRONAME:hotel.booking")

while True:

print("\n1. Book a Room\n2. Cancel Booking\n3. Exit")

choice = input("Enter your choice: ")

if choice == "1":

 guest_name = input("Enter guest name: ")

 print(hotel.book_room(guest_name))

elif choice == "2":

 guest_name = input("Enter guest name: ")

 print(hotel.cancel_booking(guest_name))

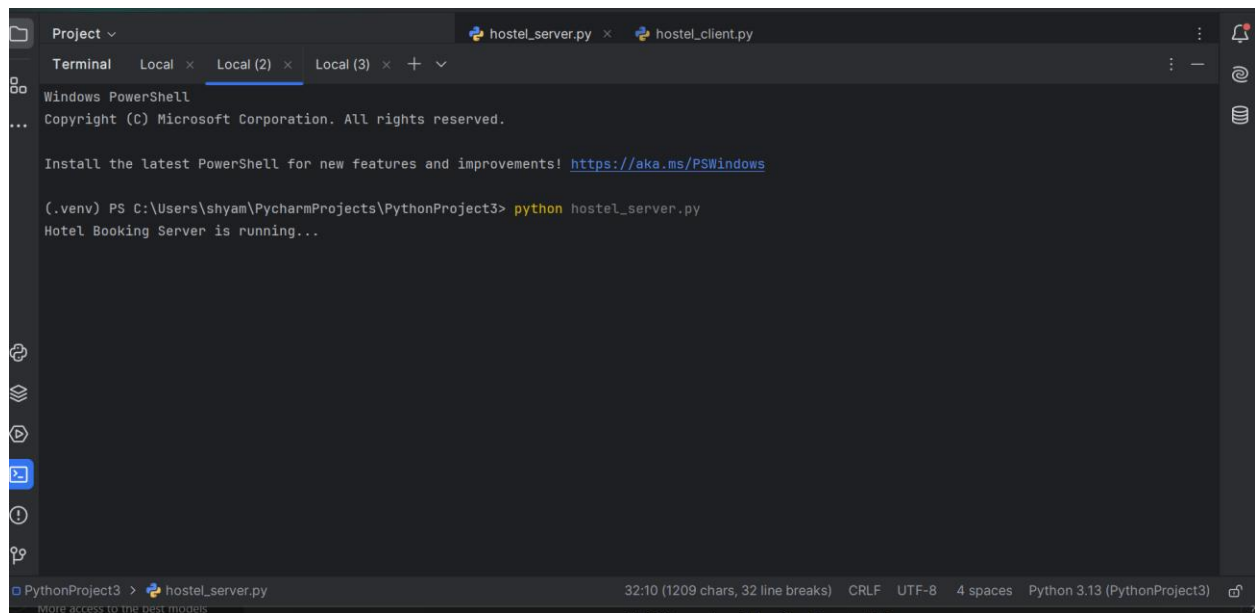
elif choice == "3":

 print("Exiting...")

 break

else:

 print("Invalid choice! Please try again.")

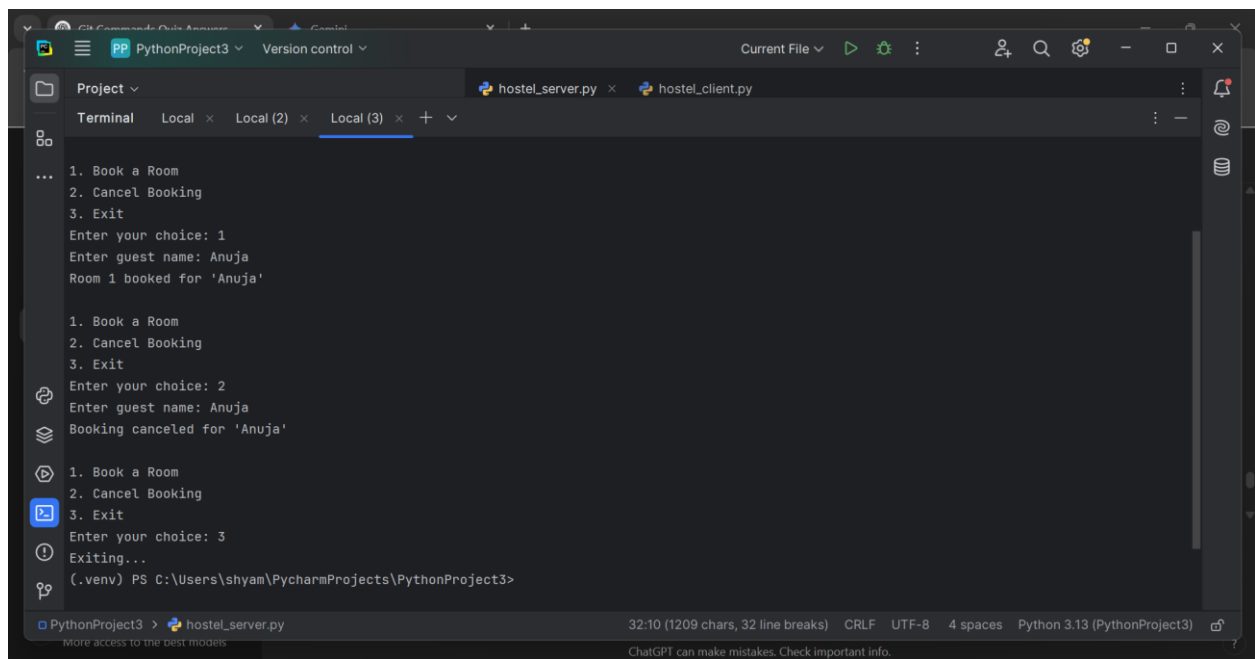


The image shows a PyCharm IDE window with a terminal pane at the bottom. The terminal is running a Windows PowerShell session. It displays the standard PowerShell copyright notice and a message to install the latest PowerShell. The user has executed the command `python hostel_server.py` in a virtual environment, and the output is "Hotel Booking Server is running...". The terminal tabs show "Local (2)" is selected. The file explorer on the left shows the project structure, and the bottom status bar indicates the file is `hostel_server.py` in a Python 3.13 environment.

```
Project: PythonProject3
Terminal: Local (2)
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

(.venv) PS C:\Users\shyam\PycharmProjects\PythonProject3> python hostel_server.py
Hotel Booking Server is running...
```



The image shows the same PyCharm IDE window, but the terminal now shows the interactive output of the `hostel_server.py` script. The script presents a menu with three options: "1. Book a Room", "2. Cancel Booking", and "3. Exit". The user has entered "1" to book a room, then "Anuja" as the guest name, resulting in "Room 1 booked for 'Anuja'". The user then entered "2" to cancel the booking, resulting in "Booking canceled for 'Anuja'". Finally, the user entered "3" to exit, and the script outputs "Exiting...". The terminal tabs now show "Local (3)" is selected. The bottom status bar remains the same, indicating the file is `hostel_server.py` in a Python 3.13 environment.

```
Project: PythonProject3
Terminal: Local (3)
...
1. Book a Room
2. Cancel Booking
3. Exit
Enter your choice: 1
Enter guest name: Anuja
Room 1 booked for 'Anuja'

1. Book a Room
2. Cancel Booking
3. Exit
Enter your choice: 2
Enter guest name: Anuja
Booking canceled for 'Anuja'

1. Book a Room
2. Cancel Booking
3. Exit
Enter your choice: 3
Exiting...
(.venv) PS C:\Users\shyam\PycharmProjects\PythonProject3>
```